REMARKS

The Board of Patent Appeals and Interferences is thanked for rendering the Decision on Appeal in this case.

Paragraphs [11], [13], [23], [25], [35] and [39] in the applicants' specification are amended. Claims 1, 5-6, 9, 11, 15, 17, 19-21, 24, 27 and 29 -34 are amended. Claims 2 and 16 are canceled. Claims 3-4, 7-8, 10, 12-14, 18, and 22 were previously canceled. Hence, Claims 1, 5-6, 9, 11, 15, 17, 19-21, and 23-34 remain pending in the application.

CLAIM REJECTIONS UNDER 35 U.S.C. §103(a)

Claims 1-2, 5-6, 9-11, 15-17, 19-21 and 23-34 are rejected under 35 U.S.C. §103(a) as allegedly unpatentable over Agrawal et al., U.S. Patent Application No. 2002/0004813 A1 ("Agrawal") in view of Claussen et al., U.S. Patent No. 6,636,863 ("Claussen"). (Final Office Action: page 2) The rejection is respectfully traversed.

CLAIM 1

Among other features, present Claim 1 recites:

analyzing a page that includes static markup text and a set of code instructions executable on a server;
extracting the static markup text from the page **and storing the static markup text in a resource file;**
generating a servlet class for the page based on the set of code instructions, **wherein the servlet class comprises a static initializer for initializing a static class variable for the servlet class,** and wherein the servlet class does not include the static markup text;
in response to a first use of the servlet class by any instance of the servlet class:
invoking the static class initializer of the servlet class, wherein the invoking of the static class initializer of the servlet class causes the static markup text to be read from the resource file, and the static class variable to be initialized with the static markup text;
loading a copy of the static class variable initialized with the static markup text into shared memory;
in response to each request of a plurality of requests for the page from a plurality of clients, performing the steps of:
instantiating a distinct instance of the servlet class on the server, wherein instantiating each instance of the servlet class does not create another copy of the static markup text;

executing said distinct instance of the servlet class, wherein execution of each instance of the server class generates a compiled page **by combining the static markup text that resides in the shared memory with results produced by executing the set of code instructions;** and
sending the compiled page to a client that requested the page;

Support for the amendment is provided at least in paragraphs [25] – [27], [30], [11] and [36]-[37] of the applicants' specification.

It is well founded that to establish a *prima facie* case of obviousness under 35 U.S.C. §103(a), the references cited and relied upon must teach or suggest all the claim limitations. In addition, a sufficient factual basis to support the obviousness rejection must be proffered. *In re Freed*, 165 USPQ 570 (CCPA 1970); *In re Warner*, 154 USPQ 173 (CCPA 1967); *In re Lunsford*, 148 USPQ 721 (CCPA 1966).

Agrawal does not describe that generating and storing a block in a cache involves **reading any static markup text from a resource file** and **using the read static markup text to initialize the static class variable of any instantiation of the servlet class**, as claimed. In Agrawal, when a web page is requested, the blocks are retrieved from the cache and used to assemble the requested page. (Agrawal: Para [39]) Retrieving a particular block from the cache causes executing the code included in the block and accessing a data source in which data needed to generate a corresponding portion of the page are stored. However, Agrawal does not describe that a static class initializer of any servlet class is invoked in response to a first use of the servlet class to cause static markup text to be read from a resource file, in which the static markup text extracted from the web page has been stored, as claimed.

Agrawal does not initialize static class variables for the servlet class **when any instance of the servlet class is used for the first time**, as claimed. Agrawal generates and stores the whole block (or blocks) in the cache. (Agrawal: Para [36]-[39]) However, Agrawal does not invoke a static class initializer of a servlet class when the servlet class is used for the first time by any instance of the servlet class to read from a resource file a static markup text and to initialize

the static class variable with the static markup text read from the resource file, as claimed.

Claussen does not cure the deficiencies of Agrawal. Claussen translates a requested page file into a Java servlet, compiles a servlet, class loads the servlet, and then invokes the servlet to cause a given web content to be returned to the requesting browser. (Claussen: Col. 2, ll. 13-20) However, just as Agrawal, **Claussen does not initialize static class variables for the servlet class with static markup text read from a resource file when any instance of the servlet class is used for the first time,** as claimed.

Further, Agrawal and Claussen, individually or in combination, fail to describe:

> **loading a copy of the static class variable initialized with the static markup text into shared memory;**

Agrawal loads to a cache blocks which contain code and references to a data store, but does not load to shared memory a copy of a static class variable that has been already initialized with static markup text read from a resource file, as claimed. Agrawal's block does not have static class variables that have been already initialized with static markup text, as claimed.

Claussen does not cure the deficiencies of Agrawal because Claussen invokes a servlet to generate the entire web page, **not to load a copy of the static class variable initialized with static markup text into shared memory,** as claimed. Claussen does not describe that a copy of static class variable is initialized with the static markup text and loaded into the shared memory, as claimed.

Moreover, Agrawal and Claussen, individually or in combination, fail to describe:

> executing said distinct instance of the servlet class, wherein execution of each instance of the server class generates a compiled page **by combining the static markup text that resides in the shared memory with results produced by executing the set of code instructions;**

Agrawal executes an instance of a servlet class to retrieve a cached block from a cache

and generates a portion of a compiled page using the code included in the block and data stored in a data store. However, the data stored in the data store is **not static markup text stored in shared memory, and used to initialize static class variables,** as claimed.

In Agrawal, the results produced by executing code included in a block constitute a respective portion of a web page and the portion is combined with other portions generated when the code of other blocks is executed. (Agrawal: Para [56]) However, none of the results produced in Agrawal is combined with static markup text that resides in the shared memory, as claimed. None of the results produced in Agrawal is combined with the static markup text that was placed in the shared memory when static class variables were initialized using a static class initializer of the servlet class, as claimed.

Claussen does not cure the deficiencies of Agrawal because Claussen is silent about **combining results produced by executing code instructions with static markup text that is stored in shared memory when static class variables of a servlet class were initialized**, as claimed.

Therefore, Agrawal and Claussen, individually or in combination, fail to teach or suggest one or more features recited in Claim 1. Therefore, Claim 1 is patentable over Agrawal and Claussen. Reconsideration and withdrawal of the rejection is respectfully requested.

CLAIMS 5, 15 AND 19

Each of Claims 5, 15 and 19 recites features similar to those in Claim 1. Therefore, each of Claims 5, 15 and 19 is patentable over Agrawal and Claussen for the same reasons as for Claim 1. Reconsideration and withdrawal of the rejection are respectfully requested.

DEPENDENT CLAIMS

The claims that are not discussed above depend directly or indirectly on the claims that have been discussed. Therefore, those claims are patentable for the reasons given above. In addition, each of the dependent claims separately introduces features that independently render the claim patentable. However, due to the fundamental differences already identified, and to expedite positive resolution of the examination, separate arguments are not provided for each of

the dependent claims at this time.

CONCLUSION

For the reasons set forth above, all pending claims are in condition for allowance. A petition for an extension of time is hereby made to the extent necessary to make this reply timely filed. If any applicable fee is missing or insufficient, the Commissioner is authorized to charge any applicable fee to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Date: May 13, 2010

/MalgorzataAKulczycka#50496/
Malgorzata A. Kulczycka
Reg. No. 50,496

2055 Gateway Place, Suite 550
San Jose, California 95110
Telephone: (408) 414-1228
Facsimile: (408) 414-1076